Automatic Retrieval of Parallel Collocations

Valeriy I. Novitskiy

The Moscow Institute of Physics and Technology, Moscow, Russia
nov.valerij@gmail.com

Abstract. An approach to automatic retrieval of parallel (two-language) collocations is described. The method is based on comparison of syntactic trees of two parallel sentences. The key feature of the method is a sequence of filters for getting more precise results.

Keywords: NLP, parallel collocations, automatic information extraction, text mining.

1 Introduction

Most of natural languages consist of hundreds of thousands of words. The amount of two-word combinations is $\approx 10^{10}$, but only a few of them are real collocations.

In this work we study the problem of extracting parallel collocations. A parallel collocation is a combination of a collocation and its translation into another language. We are interested in non-trivial literal translations. Parallel collocations make a valuable linguistic resource. For example, they can be used as an auxiliary material by linguists or as a statistical data in different NLP tasks.

The key feature of the approach described in this work is a sequence of heuristic filters which help to extract the most valuable collocations. Extraction of every possible collocation is intractable, no precise algorithm for the solution of this problem is known. Hence, we have to use some simplifications coming to soft computing of collocations.

Suppose we are given a corpus of parallel texts¹. These texts are aligned sentence-to-sentence, which means that we know the matchings between parallel units of texts (usually a text unit is one sentence). We use a syntactic analyzer to parse the texts and return respective syntactic trees.

The work described in this paper had several goals:

1. Designing an algorithm for automatic collocation retrieval (with some specific restrictions described below).
2. Collecting statistical data to improve the work of syntactic analyzer in use.
3. Improving two-language dictionary by adding new translations.
4. Creating a Translation Memory database of parallel collocations which can be used as a reference-book by linguists.

¹ Texts and its translations into different language.

1.1 Environment

In our work we used several tools developed in ABBYY:

1. English-Russian dictionary
2. Syntactic analyzer
3. Word-to-Word matching algorithm used for sentence aligning.

The dictionary is based on semantic invariants (or *classes*). For every language there are several possible realizations of those classes (e.g., “competition”, “gala” and “event” may be placed in one “competition” class). At the same time homonyms are placed in several classes simultaneously. The corporate dictionary is comprehensive enough and consists of more than 60 000 classes. Distinguishing between homonyms (disambiguation) is done during text analysis and is not discussed here.

The syntactic analyzer takes a single sentence and outputs the best syntactic tree² based on internal estimations of quality. The nodes of the tree are semantic classes and arcs are connections between words. It is possible to get wrong results (incorrect words sense selection). In this case we suppose that either wrong collocations will not be produced at all (due to differences of syntactic trees in different languages) or their frequency will be rather small to delete them by the following filtration procedures. The refinement of syntactic tree is attained by parallel analysis of sentences in both languages.

This work does not use such well-known criteria as Mutual Information [2], t-score measure or log-likelihood measure [3], because we choose more predictable selection algorithms. There are many purely statistical algorithms of collocation extraction based on words cooccurrence (e.g., see [4], [5]), but they do not use information about connections between words which we are interested in.

In contrast to other works where syntactic information from texts is used [6], we do not restrict our search to two-words collocations, but look for collocations of different length. In this case one encounters the problem of “partial” collocations: partial collocations are parts of some larger collocations. Here we introduce a new approach to discarding partial collocations.

2 Description of the Method

The algorithm for retrieving collocations can be divided into the following steps:

1. Word-to-word sentence aligning.
2. Single language collocations generation.
3. Matching collocations of different languages to compose parallel collocations.
4. Filtration of infrequent and occasional collocations.

Below we consider each step of the approach in detail.

² Different representation of sentence structure can be found in [1].

2.1 Word-to-Word Sentence Alignment

We use a dictionary based on semantic invariants, which means that all possible synonyms of a word-sense are placed in one *semantic class*. So the aligning process has to solve the following two main problems:

- Homonyms matching (due to possible incorrect semantic variant choice).
- Several synonyms in one sentence.

The first problem can be solved rather easily. We can take all possible semantic classes to which a word can belong and compare them with all possible classes of the opposite word (pic. 1). In the case they do not have empty intersection we can consider them as translations of each other.

Fig. 1. Semantic classes intersection

The solution of the second problem is based on syntactic trees. We take into account dependencies between words in sentence. For example, we can match two words with better confidence if their parents in syntactic tree correspond to each other.

We compare all words between two sentences and estimate the “quality” of each pair (as a number). Impossible pairs are suppressed by setting prohibition penalty. Then we search for the best pairs (by their integral quality). Thus the problem is reduced to the well-known problem of searching for the best matching in a bipartite graph. This problem has a polynomial-time solution algorithm, e.g. a so-called “Hungarian algorithm” (see [7]).

2.2 One-Language Collocation Production

Let us impose the following constraints on collocations:

- The number of words in a collocation ranges from one to five.
- The result is a subtree of the syntactic tree.
- There are no pronouns in collocations.
- Syntactic word³ can not be a root of collocation subtree.
- We allow only one “gap” of limited size in the linear realization of a collocation in a sentence.

³ Like pronoun, preposition, auxiliary verb and so on.

2.3 Parallel Collocations Production

We use the results of two previous steps, namely the alignment of parallel sentences and the set of variants of one-language collocations from the same sentences. This information allows us to select one-language collocations and produce candidates for parallel collocations. We impose the following constraints:

- The length difference between a collocation and its translation is less or equal to one word.
- There are some word-to-word correspondences between collocations (the longer collocation the more correspondences there should be).
- It is possible to have no correspondences for short (1–2 words length) collocations. Instead, there should be correspondences between collocation roots and all of their children in the syntactic tree.
- There are no “outgoing” correspondences (that “go out” from a collocation but do not “come into” its translation).

During this step all possible collocations are produced. For example, in the corpus of ≈ 4 bln fragments there are more than 100 bln different collocations. But only 7 bln of them occur twice as frequent and more.

2.4 Filtration

At this step we select only valuable collocations from that variety we got during the previous step. The main idea is that stable collocations are rather frequent and (almost) always have the same translation. We will omit collocations that are infrequent and have different translations. Some other heuristics will be uncovered below.

There are several collocation filters:

1. Removal of rare collocations (preliminary filtration by frequency with lower threshold).
2. Removal of collocations with stop words.
3. Removal of inner collocations, i.e., those that are parts of another. For example, “Organization for Security” is an internal part of “Organization for Security and Co-operation in Europe”.
4. Similarly we should remove “outer” collocations that can occasionally appear. For example, “in the United Nations” is the outer collocations to the just “United Nations”.
5. Selection of one translation for each collocations (if it is not too ambiguous). We leave translation if it appears not less than in 70% cases.
6. Final removal of rare collocations.
7. Removal of well-known translations (found in the dictionary).

3 Discussion of Results

3.1 Explanation of Filters Order

Our experiments showed that the proposed sequence of filters results in the highest precision possible with these filters. The filters are applied step by step as they are listed above. Let us explain the reason for such ordering. The frequency filter discards rare collocations that are not interesting for further study. We consider that we cannot prove significance of a rare collocation by any of the following statistical examinations. The stop word filter discards a priori noninteresting variants, reducing the workload on the following steps. The important fact is these two filters remove more than 95% of collocations. Hence, other steps are performed faster and with more precision. Collocations are usually generated with their “sub” and “super” parts. The next filter aims at narrowing such a “range” of collocation (that have one-word difference with each other). Ideally it should leave only one collocation from this “range”. Close collocations are compared to each other. Such comparison is sensitive to the “gaps” (absence of collocations with one changed word). That is why this filter is applied at the first steps.

Disambiguation of collocations translation (selection one main variant of translation) causes such “gaps”. This is a reason why this filter is used after “sub-super” collocation filtration step. At the same time the previous two filters deal with single-language parts of collocations and disambiguation filter deals with both parts. It is significant to eliminate as many wrong rare variants as possible before this step. For example, if we have two variants of translation “старый дверной замок”: “old door-lock” (right) and “door-lock”(partial, wrong) this filter would eliminate both of them (there are no dominant variant). But in fact wrong variant is removed on the previous step.

The second step of filtration by frequency removes those infrequent collocations that supported the performance of “inner-outer” and “disambiguation” filters. With the filter of word-to-word translations they remove collocations we are not interested in (rare or well-known translations). They can be run in any order with the same result.

3.2 Results of Filtration

Experiments were carried out on an English-Russian corpus which consists of about $4.2 \cdot 10^6$ parallel sentences.

There are $\approx 62 \cdot 10^6$ unique pairs of collocation and their translations after the step of “Parallel Collocations Generation”. Most of them ($\approx 56 \cdot 10^6$) occur only once in corpus. Filtration results are shown in Fig. 2.

As a result we obtain about $42.5 \cdot 10^3$ of parallel collocations. The result may seem rather modest from the first glance, but it can be significantly improved by adjusting filters’ thresholds (in particular, of the “Ambiguity translations” filter) and increasing the number of texts in the collection. Several examples are shown in Fig. 3.

Filter	Output
By frequency (preliminary)	2.5 bln.
By stop-word list	1.1 bln.
“Inner” and “outer” collocations	≈ 568 000
Ambiguity translations	≈ 105 000
Translated by dictionary	≈ 66 500
By frequency (finally)	42 636

Fig. 2. Filtration process

English	Russian	Occurrences
job time	срок задания	12
galaxy space	космическое пространство	13
other foreign object	иной посторонний объект	5
air transport field	область воздушного транспорта	30
to be beyond the scope of book	выходить за рамки книги	29
to establish under article	учреждать в соответствии со статьёй	75

Fig. 3. Examples of extracted parallel collocations

There are two main measures of results quality: precision and recall. In this work the precision is much more important than the recall. The reason is that it is very difficult to inspect tens of thousand collocation to found ones with errors. The recall, as the amount of collocations generated, can be made larger with the growth of the text base. We prefer to omit rare collocations by analyzing matching errors, however statistical methods like Mutual Information can select them as rare and unexpected word combinations. The precision in our research computed by using collocations for analysis of test corpora. We compare results of analysis with and without collocations. Any found errors are analyzed and the general algorithm is updated to avoid them.

An open problem is how an efficient estimate of precision and recall. Manual markup of large text bases is almost unfeasible. We estimate precision by comparing results with opinions of a random subset of experts. Another problem is comparing with statistical algorithms. The comparison is connected with the difference in produced information (the existence of syntactic links between words in statistical collocations). However, the manual check of random collocations shows good quality in general. An example of such precision estimation is shown in Fig. 4. There are two categories of not good collocations that can be refined by either improving dictionary entities or by customizing the algorithm. An improvement of the algorithm may be attained by eliminating duplicates of text fragments (that can appear in manuals, government documents and so on). We can achieve precision of more than 80% on this sample by introducing this technique.

Quality	Percent
Good collocations	67
Improvable with dictionary	4
Improvable with algorithm	16
Others	12

Fig. 4. Result of manual checking of a random subset of 100 collocations

4 Conclusion

The main result of our work is a method which proved to be useful and is employed now in software for collocation search.

There are several possible way to improve the proposed method:

- Introducing a quality measure for collocations (for ranking them and selecting the best ones).
- Tuning filters thresholds.
- Improving corpora used in computations (by correcting spelling errors and removing occasional bad parallel fragments).

References

1. Bolshakov, I.A.: Computational Linguistics: Models, Resources, Applications. In: Bolshakov, I.A., Gelbukh, A.F. (eds.) IPN - UNAM - Fondo de Cultura Economica (2004)
2. Church, K.W.: Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1), 22–29 (1990)
3. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1), 61–74 (1993)
4. Smadja, F.A.: Retrieving collocations from text: Xtract. *Computational Linguistics* 19(1), 143–177 (1993)
5. Bouma, G.: Collocation extraction beyond the independence assumption. In: *Proceedings of the ACL 2010 Conference Short Papers. ACLShort 2010*, pp. 109–114. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
6. Evert, S.: The Statistics of Word Cooccurrences Word Pairs and Collocations. Ph.D. thesis / Universität Stuttgart. Institut für Maschinelle Sprachverarbeitung (IMS) (2004)
7. Burkard, R.: *Assignment Problems*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia (2009)